We claim:

1.   A method of processing an application request on an end user application and an application server comprising the steps of:

5

a)   initiating the application request on the end user application in a first language with a first application program;

b)   transmitting the application request to the server and converting the application
10   request from the first language of the first end user application to a language running on the application server;

c)   processing said application request on the application server;

15   d)   transmitting a response to the application request from the application server to the end user application, and converting theresponse to the application request from the language running on the application server to the first language of the first end user application; and

20   e)   wherein the end user application and the application server have at least one connector therebetween, and the steps of (i) converting the application request from the first language of the first end user application as a source language to the language running on the application server as a target language, and (ii) converting a response to the application request from the language running on the
25   application server as a source language to the first language of the first end user application as a target language, each comprise the steps of:

1)   invoking connector metamodels and type descriptor metamodels of respective source and target languages;

30

2) populating the connector metamodels with metamodel data of each
of the respective source and target languages; and

3) converting the source language to the target language.

5

2. The method of claim 1 wherein the end user application is a web browser.

3. The method of claim 2 wherein the end user application is connected to the
application server through a web server, and the web server comprises an connector.

10

4. The method of claim 1 wherein the metamodel datacomprises invocation
metamodel metadata, application domain interface metamodel metadata, and type
descriptor metamodel metadata.

15   5. The method of claim 4 wherein the type descriptor metamodel datadefines
physical realizations, storage mapping, data types, data structures, and realization
constraints.

6. A transaction processing system comprising a client, a server, and at least one
20   connector therebetween,

a) the client having an end user application, and being controlled and configured to
initiate an application request with the server in a first language with a first
application program and to transmit the application request to the server;

25

b) the connector being configured and controlled to receive the application request
from the client, convert the application request from the first language of the first
end user application running on the client to a language running on the server;

30   c) the server being configured and controlled to receive the converted application
request from the connector and processing the said application request in a second

language with a second application program residing on the server, and to thereafter transmit a response to the application request through the connector back to the first application program on the client;

5   d)   the connector being configured and controlled to receive a response to the application request from the server, to convert a response to the application request from the language running on the application server to the first language of the first application program running on the client; and

10   e)   wherein connector between the client and the server is configured and controlled to (i) convert the application request from the first language of the client application on the client as a source language to the language running on the application server as a target language, and (ii) convert the response to the application request from the language running on the application server as a source language to the first language of the client application running on the client as a target language, each by a method comprising the steps of:

           1)   retrieving connector metamodels of respective source and target languages from a metamodel datarepository;

20

           2)   populating the connector metamodels with metamodel data and type descriptor metamodel from the metamodel datarepository for each of the respective source and target languages; and

25   3)   invoking the retrieved, populated connector metamodels and converting the source language to the target language.

7.   The system of claim 6 wherein the end user application is a web browser.

30   8.   The system of claim 7 wherein the end user application is connected to the application server through a web server, and the web server comprises an connector.

SVL920010038US1

9.   The system of claim 6 wherein the metamodel datacomprises invocation metamodel metadata, application domain interface metamodel metadata, and type descriptor metamodel metadata.

5

10.   The system of claim 9 wherein the type descriptor metamodel datadefines physical realizations, storage mapping, data types, data structures, and realization constraints.

10   11.   A transaction processing system configured and controlled to interact with a client application, and comprising a server, and at least one connector between the server and the client application, where the client has an end user application, and is controlled and configured to initiate an application request with the server in a first language with a first application program and to transmit theapplication request to the server, wherein:

15

       a)   the connector being configured and controlled to receive an application request from the client, convert the application request from the first language of the first end user application running on the client to a language running on the server;

20   b)   the server being configured and controlled to receive the converted application request from the connector and process the said application request in a second language with a second application program residing on the server, and to thereafter transmit a response to the application request through the connector back to the first application program on the client;

25

       c)   the connector being configured and controlled to receive the application request from the server, to convert a response to the application request from the language running on the application server to the first language of the first application program running on the client; and

30

61

d)   wherein connector between the client and the server is configured and controlled to (i) convert the application request from the first language of the client application on the client as a source language to the language running on the application server as a target language, and (ii) convert the response to the application request from the language running on the application server as a source language to the first language of the client application running on the client as a target language, each by a method comprising the steps of:

       1)   retrieving connector metamodel dataof respective source and target languages from a metamodel datarepository;

       2)   populating the connector metamodels with metamodel data and type descriptor metamodel data of each of the respective source and target languages from the metamodel datarepository and invoking the retrieved, populated connector metamodels; and

       3)   converting the source language to the target language.

12.   The system of claim 11 wherein the end user application is a web browser.

13.   The system of claim 12 wherein the end user application is connected to the application server through a web server, and the web server comprises an connector.

14.   The system of claim 11 wherein the metamodel datacomprises invocation metamodel metadata, application domain interface metamodel metadata, and type descriptor metamodel metadata.

15.   The system of claim 14 wherein the type descriptor metamodel datadefines physical realizations, storage mapping, data types, data structures, and realization constraints.

16.   A groupware system having a plurality of end user applications, said end user applications each comprising an e-mail client, a content database client, and a content replication client, said system further comprising an e-mail server, a content database server, and a content replication server, said groupware system being configured and

5   controlled to communicate between disparate end user applications, said groupware system comprising at least one connector between a server and an end user application, wherein the end user application is controlled and configured to participate with a server in a first language with a first application program and the server is configured and controlled to participate with the client in a second language, and wherein:

10

a)   the connector is configured and controlled to receive an application request from the end user application, convert the application request from the first language of the first end user application to a language running on the server;

15   b)   the server being configured and controlled to receive the converted application request from the connector and process the said application request in a second language with a second application program residing on the server, and to thereafter transmit the response to the application request through the connector back to the end user application;

20

c)   the connector being configured and controlled to receive the response to the application request from the server, to convert a response to the application request from the language running on the application server to the first language of the end user application; and

25

d)   wherein connector between the end user application program and the server is configured and controlled to (i) convert the application request from the first language of the end user application as a source language to the language running on the application server as a target language, and (ii) convert the response to the

30   application request from the language running on the application server as a

source language to the language of the end user application as a target language, each by a method comprising the steps of:

> 1) retrieving connector metamodel dataof respective source and target languages from a metamodel datarepository;

> 2) populating the connector metamodels with metamodel data and type descriptor metamodel data of each of the respective source and target languages from the metamodel datarepository and invoking the retrieved, populated connector metamodels; and

> 3) converting the source language to the target language.

17. The groupware system of claim 16 wherein the metamodel datacomprises invocation metamodel metadata, application domain interface metamodel metadata, and type descriptor metamodel metadata.

18. The system of claim 17 wherein the type descriptor metamodel datadefines physical realizations, storage mapping, data types, data structures, and realization constraints.

19. A program product comprising a storage medium having invocation metamodel metadata, application domain interface metamodel metadata, language metamodel metadata, and type descriptor metamodel data and computer instructions for building a metamodel datarepository of source and target language metamodel metadata.

20. The program product of claim 19 further comprising computer instructions for building connector stubs from said metamodel metadata.

21. The program product of claim 19 further comprising computer instructions to build a connector for carrying out the steps of:

SVL920010038US1

1) retrieving connector metamodel dataof respective source and target languages from the metamodel datarepository;

5 2) populating the connector metamodels with metamodel data of each of the respective source and target languages from the metamodel datarepository and invoking the retrieved, populated connector metamodels; and

10 3) converting the source language to the target language.

22. The program product of claim 21 wherein the metamodel datain the repository comprises invocation metamodel metadata, application domain interface metamodel metadata, and type descriptor metamodel metadata.

15

23. The program product of claim 22 wherein the invocation metamodel datais chosen from the group consisting of message control information, security data, transactional semantics, trace and debug information, pre-condition and post-condition resources, and user data.

20

24. The program product of claim 22 wherein the application domain interface metamodel datacomprises input parameter signatures, output parameter signatures, and return types.

25 25, The program product of claim 22 wherein the type descriptor metamodel datadefines physical realizations, storage mapping, data types, data structures, and realization constraints.